This listing of claims will replace all prior versions, and listings, of claims in the application:

<u>Listing of Claims</u>:

Claim 1 (Currently Amended): A method for optimising compiler-generated program code intended for a portable data carrier having a processor core and a first and second memory area, comprising:

the first memory area being provided to receive the optimised program code,

the second memory area being provided to receive a predefined library having a multiplicity of library code fragments, <u>wherein the predefined library is independent of the compiler-generated program code that is to be optimized,</u> and

the compiler-generated program code being searched for program code fragments ~~which correspond, at least in respect of their effect, to respectively one~~ <u>that perform the same function as a respective one of the</u> library code ~~fragment~~ <u>fragments</u>, the program code fragments found thereby being replaced by respectively one call of the corresponding library code fragment.

Claim 2 (Previously Presented): The method according to claim 1, wherein a program code fragment is replaced by a library code fragment only if both code fragments are identical in their form as executable machine code.

Claim 3 (Previously Presented): The method according to claim 1, wherein at least some library code fragments are parameterised.

Claims 4-10 (Canceled).

Claim 11 (Previously Presented): The method according to claim 1, wherein a program code fragment to be replaced is replaced, at least if the program code fragment does not interfere with the program flow, by a subroutine call instruction to the corresponding library code fragment.

Claim 12 (Previously Presented): The method according to claim 1, wherein the compiler-generated program code exists in the form of assembler source code, and the optimization procedure is performed on a source code level.

Claim 13 (Previously Presented): The method according to claim 1, wherein the predefined library is matched to at least one of the following:

 the hardware of the portable data carrier,

 an operating system of the portable data carrier, and

 a compiler used in the generation of the compiler-generated program code.

Claim 14 (Previously Presented): The method according to claim 1, wherein the first memory area is electrically programmable.

Claim 15 (Previously Presented): The method according to claim 1, wherein the second memory area is mask-programmable.

Claim 16 (Previously Presented): The method according to claim 1, wherein the first memory area occupies more chip area per memory cell in the portable data carrier than is occupied by the second memory area.

Claim 17 (Currently Amended): A computer program product comprising <u>a computer-readable storage medium and</u> program instructions for a general-purpose computer <u>stored in the computer-readable storage medium, the program instructions causing</u> ~~which cause~~ the general-purpose computer to optimize compiler-generated program code intended for a portable data carrier having both a processor ~~core~~ and a first and second memory area, the first memory area being provided to receive the optimized program code, the second memory area being provided to receive a predefined library having a multiplicity of library code fragments, wherein <u>the predefined library is independent of the compiler-generated program code that is to be optimized, and</u>

 the optimization includes searching the compiler-generated program code for program code fragments ~~which correspond, at least in respect of their effect, to respectively one~~ <u>that perform the same function as a respective one of the</u> library code ~~fragment~~ <u>fragments</u>, the program code fragments found thereby being replaced by respectively one call of the corresponding library code fragment.

Claim 18 (Previously Presented): The computer program product according to claim 17, wherein the program instructions additionally implement a compiler for converting a high-level language source code into the compiler-generated program code.

Claim 19 (Previously Presented): The computer program product according to claim 17, wherein a program code fragment is replaced by a library code fragment only if both code fragments are identical in their form as executable machine code.

Claim 20 (Previously Presented): The computer program product according to claim 17, wherein a program code fragment to be replaced is replaced, at least if the program code fragment does not interfere with the program flow, by a subroutine call instruction to the corresponding library code fragment.

Claim 21 (Previously Presented): The computer program product according to claim 17, wherein the compiler-generated program code exists in the form of assembler source code, and the optimization procedure is performed on a source code level.

Claim 22 (Previously Presented): The computer program product according to claim 17, wherein the predefined library is matched to at least one of the following:

 the hardware of the portable data carrier,

 an operating system of the portable data carrier, and

 a compiler used in the generation of the compiler-generated program code.

Claim 23 (Previously Presented): The computer program product according to claim 17, wherein the first memory area is electrically programmable, and the second memory area is mask-programmable, and the first memory area occupies more chip area per memory cell in the portable data carrier than is occupied by the second memory area.

Claim 24 (Currently Amended): A portable data carrier having a processor ~~core~~, a first memory area and a second memory area, there being contained in the first memory area optimized program code, and there being contained in the second memory area a library which is predefined independently of the optimized program code and has a multiplicity of library code fragments, <u>wherein the predefined library is independent of the compiler-generated program code that is to be optimized, and</u> wherein the optimized program code has been obtained from compiler-generated program code by searching for program code fragments ~~which correspond, at least in respect of their effect, to respectively one~~ <u>that perform the same function as a respective one of the</u> library code ~~fragment~~ <u>fragments</u>, the program code fragments found thereby being replaced by respectively one call of the corresponding library code fragment.

Claim 25 (Previously Presented): The portable data carrier according to claim 24, wherein, when obtaining the optimized program code from the compiler-generated program code, a program code fragment is replaced by a library code fragment only if both code fragments are identical in their form as executable machine code.

Claim 26 (Previously Presented): The portable data carrier according to claim 24, wherein the predefined library is matched to at least one of the following:

      the hardware of the portable data carrier,

      an operating system of the portable data carrier, and

      a compiler used in the generation of the compiler-generated program code.

Claim 27 (Previously Presented): The portable data carrier according to claim 24, wherein the first memory area is electrically programmable, and the second memory area is mask-programmable, and the first memory area occupies more chip area per memory cell in the portable data carrier than is occupied by the second memory area.